



# UNITED STATES PATENT AND TRADEMARK OFFICE

mn

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/646,289	08/21/2003	Son Ho	MP0390.1	9390
26703 7590 05/07/2007 HARNESSE, DICKY & PIERCE P.L.C. 5445 CORPORATE DRIVE SUITE 200 TROY, MI 48098			EXAMINER PATEL, KAUSHIKKUMAR M	
			ART UNIT 2188	PAPER NUMBER
			MAIL DATE 05/07/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/646,289	<b>Applicant(s)</b> HO ET AL.	
	<b>Examiner</b> Kaushikkumar Patel	<b>Art Unit</b> 2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 05 March 2007.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-2, 4-7, 9-10, 12-15 and 17-33 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1, 2, 4-7, 9, 10, 12-15 and 17-33 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 21 August 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date <u>1/4/07, 1/11/07</u> . | 6) <input type="checkbox"/> Other: _____  |

## **DETAILED ACTION**

### ***Response to Amendment***

1. This office action is in response to applicant's communication filed March 05, 2007 in response to PTO office action mailed December 05, 2006. The applicant's remarks and amendments to the claims and/or specification were considered with the results that follow.
2. In response to the last office action, claims 1, 5, 10, 13, 18 and 26 have been amended. No claims have been added. No claims have been canceled. As a result, claims 1-2, 4-7, 9-10, 12-15 and 17-33 remain pending in this application.
3. Double patenting rejections of claims had been withdrawn due to Terminal Disclaimer filed on March 05, 2007. The claim objections are also withdrawn due to applicant's remarks (page 15) filed on March 05, 2007.

### ***Response to Arguments***

4. Applicant's with respect to claims 1, 5, 13, 18 and 26 have been considered but are moot in view of the new ground(s) of rejection.

### ***Information Disclosure Statement***

5. The information disclosure statement (IDS) submitted on January 4 and 11, 2007 had considered by the examiner.

***Admitted Prior Art***

6. Applicant has not traversed the Examiner's assertion of Official Notice with regard to the rejection of claims 19 and 26 in the previous office action, therefore the well-known facts presented in these rejections are taken to be admitted prior art. These facts are summarized as follows: Computer/storage systems with cache using virtual index (virtual cache) or physical index (physical cache) as well as virtual memory addressing are well known in the art. The physical cache requires address translation for data access from cache and hence cache translating virtual to physical addressing is known in the art.

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-2, 5-7, 10, 13-15, 18-19 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zaidi et al. (US 6,601,126 B1) (Zaidi herein after), Jeddelloh (US 7,133,972 B2) and Loafman (US 2005/0021916 A1).

As per claims 1 and 18, Zaidi teaches memory storage system that is accessed by a first central processing unit (CPU) (fig. 1, item 100 and item 112, also see figs. 20-23), comprising:

a line cache including a plurality of pages that are accessed by the first CPU (fig. 1, item 126. Also caches are known to store pages accessed by CPU, see background art section of present application); and

a first memory device that stores data that is loaded into said line cache when miss occurs (fig.1, item 108 is connected to processor via cache 126 to CPU 112, column 22, lines 65-67, teaches processor use cache to access data from memory 108 and as per present application's background art section, data are loaded from lower latency memories to cache when miss occurs);

a second memory device (figs. 1-2 and 20-23, items 106, 108, 248, 250 and indicated as flash and SDRAM);

a line cache control system that controls data flow between said line cache, the first CPU, said first memory device and said second memory device (CPU accessing data from lower level storage devices through cache teaches cache control system), and that includes:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request (figs. 20-22 shows cache of CPU is connected to memory devices through MAC, which inherently teaches interfaces and CPU accessing information from cache as explained above inherently requires generating address);

a first memory interface that communicates with said first memory device and a second memory interface that communicates with said second memory device (figs. 20-23, flash and SDRAM is connected to MAC);

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache control system compares said first address to stored addresses in said line cache, returns data associated with said first address if match occurs and loads page of said line cache when miss occurs [caches are known to receive addresses from processors and comparing those addresses to stored addresses and providing data to processor if hit occurs and loading pages from lower higher latency memory in case of cache miss (see background of invention in present application)], (column 23, lines 31-34 and lines 41-45, taught as CPUs supply a request and an address, the address includes both the port, device or memory bank address and the requested memory location address. Referring to figs. 20-22, col. 23 lines 22-29, "switched channel memory controller" allows multiple DMA (and processors) to simultaneously communicate with multiple output channels. Also, col. 23, lines 40-45, suggests that CPU bus can be connected to an external flash memory through one channel and SDRAM through another channel. These statements clearly indicate that there are separate and selective communication interfaces between the devices). (switches providing separate/distinct/exclusive [limitation of claim 18] interfaces are known in the art, because a separate and independent interface avoids bus or memory bank conflict and hence increases the speed of the system, see Jeddelloh, US 7,133,972, fig. 3, col. 4, lines 30-64,).

Zaidi fails to teach loading n pages from sequential locations from memory.  
Loafman teaches when miss occurs in memory (cache); system preloads consecutive

pages from lower latency storage (Loafman, par. [0026]) into memory. It would have been obvious to one having ordinary skill in the art at the time of the invention to modify the memory storage system of Zaidi by prefetching some extra sequential pages when page fault (cache miss) occurs and the requested page is being loaded from higher latency storage to memory (cache), because during sequential access of data it is highly likely that nearby data will be accessed in near future, and by prefetching adjacent (extra) pages will avoid cache miss next time CPU references the sequential data and thus, improving the performance (see Loafman paragraphs [0011]-[0013]).

Zaidi and Loafman fail to teach loading (prefetching) n pages of cache line with the first cache miss (before second miss occurs) as required by the claim. Loafman however teaches that applications/programs read data either sequentially or randomly (also admitted by applicant, remarks submitted March 05, 2007, page 18) and if data being read randomly, prefetching may not work. He further teaches that, *"if data being read randomly and the executing program makes a request to read certain amount of data that resides on two sequential pages and if data is not already in RAM, two page faults will be raised in order to load two pages in the RAM"* (Loafman, par. [0013]). It is apparently clear from above statements that Loafman is not prefetching pages if data being read randomly, but since program requests two sequential pages (even though program accesses data randomly, it is not totally random and can access pages sequentially), two page faults will be raised (i.e. when program accessing data randomly, no prefetching occurs and hence two page faults required to load two pages), but as mentioned in par. 26, two page faults to two sequential pages, means the virtual

memory manager (VMM) infers that the data now being accessed sequentially (par. [0026], *"when program accesses two successive pages (i.e. pages 202 and 204) each using a page fault, the VMM 112 assumes that the program will continue to access data sequentially"*) and hence starts prefetching (admitted by applicant, remarks filed March 05, 2007, page 17, *"by observing the pattern used by a program"*) (Loafman, par. [0013], *"because the pages are sequential, the system may infer that the data being read sequentially; and hence, pre-fetch a block of sequential pages of data"*). Thus, it is apparently clear from above explanation, that programs read data either randomly and/or sequentially and if data being read randomly, then pre-fetching creates thrashing of cache (Loafman, pars. [0013], [0027], after reading two sequential pages using two page faults, system infers that now data is being read sequentially, hence starts pre-fetching, but the actual access is still random, so *"the blocks of pages may have been pre-fetched in vain"*), hence Loafman's system confirms that the data being read is read sequentially by having two page faults to two sequential pages before pre-fetching additional data. Thus, it is apparently clear that Loafman uses two page faults (two cache misses) to two sequential pages to confirm that data is being read sequentially. Thus, if the program reads data sequentially, then it would have been obvious to one having ordinary skill in the art at the time of the invention to start prefetching data after initial (one) miss without waiting for confirmation of whether the data being read sequentially, i.e. before a second miss occurs (as in the system of Loafman). The advantage would be to reduce number of page faults, since it is known that data is being read sequentially, there is no need for confirmation and as taught by Loafman,



prefetching works splendidly well in sequential data access and reduces page faults (Loafman, pars. [0010], [0012]), thus reducing the data access latency. Loafman teaches that as long as CPU accesses data in sequential manner, than controller keeps prefetching additional pages (2,4,8 etc. Loafman, fig. 2, items 210, 220, 230) and the next sequential page is already read ahead (pre-fetched) into the cache, hence no additional cache miss occurs.

As per claim 2, Loafman teaches that if program continue sequentially accessing prefetched pages, than prefetching more pages (four and eight and so on) into cache. (Loafman, fig. 2, paragraph [0026]).

Claims 5 and 6 are rejected under same rationales as applied to claims 1 and 2. As Loafman teaches that as long as CPU accesses data in sequential manner, than controller keeps prefetching additional pages (2,4,8 etc. Loafman, fig. 2, items 210, 220, 230) and thus next sequential address is already read ahead in the cache and hence no cache miss occurs.

As per claim 7, Loafman teaches loading 2 pages, then 4 and then 8, sequentially (Loafman, fig. 2, paragraph [0026]), which inherently teaches  $m^{\text{th}}$  page from  $n$  pages (accessing 1<sup>st</sup> then 2<sup>nd</sup> and loading 4, and then 8 pages, in case of two pages  $n = 2$  and reading 2<sup>nd</sup> page teaches reading  $m^{\text{th}}$  (2<sup>nd</sup>) page of two pages or 3<sup>rd</sup> in case of 4 pages preloaded and hence  $m$  is greater than one and less than or equal to  $n$  and prefetching 4 or 8 pages teaches additional  $n$  pages).

Claims 10 and 13-15 are rejected under same rationales as applied to claims 1-2 and 5-7 above.

As per claim 19, combination of Zaidi, Jeddeloh and Loafman teach limitations of independent claim 18, but fail to teach cache translating an address. Computer/storage systems with cache using virtual index (virtual cache) or physical index (physical cache) as well as virtual memory addressing are well known in the art. The physical caches requires address translation for data access from cache and hence cache translating virtual to physical addressing is known in the art, and examiner takes official notice of that. Physical caches are simpler to build; hence it would have been obvious to one having ordinary skill in the art at the time of the invention to use physical cache performing address translation in the system of Zaidi, Jeddeloh and Loafman.

As per claim 25, Loafman teaches loading multiple lines of data from secondary storage device to cache as explained with respect to claim 1 above.

9. Claims 4, 9, 12, 17, 26 and 33 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jeddeloh and Loafman as applied to claims 1-2, 5-7 and 13-15 above, and further in view of Barroso et al. (US 6,725,334 B2).

As per claims 4, 9, 12 and 17, Zaidi, Jeddeloh and Loafman teach a dual processor system with two caches (fig. 2, items 202 and 214, first and second processors, and items 208 and 224, two caches). Zaidi, Jeddeloh and Loafman inherently teach cache interface with first and second CPUs and both generates read requests and hence first and second address and providing an exclusive interface as taught in claim 1. Zaidi teaches system with two caches for two processors but fails to teach cache arbitration device which communicates with first and second cache

interfaces and resolves cache access conflicts between first and second CPUs. Barroso teaches a second level shared cache with switch (fig. 1, item 130 and 120), which provides interfaces with first and second CPU and arbitrates between first and second CPU (column 4, lines 10-21).

It would have been obvious to one having ordinary skill in the art at the time of invention to modify the multiple cache with multiple processor system of Zaidi and used one cache with switch as taught by Barroso to reduce the cost and the waste of the cache capacity and shared cache avoids duplication of data in individual caches (column 1, lines 45-65).

Claim 26 recites storage system with two processors and ordered data requests. Combination of claims 1 and 4, as taught above teaches dual processor system with shared cache and arbitration device. Jeddeloh teaches switch with arbitration logic to determine memory access priorities and ordering requests according to priority (Jeddeloh, col. 4, line 65 – col. 5, line 2).

As per claim 27, combination of Zaidi, Jeddeloh, Loafman and Barroso teaches limitations of independent claim 26, but fail to teach cache translating an address. Computer/storage systems with cache using virtual index (virtual cache) or physical index (physical cache) as well as virtual memory addressing are well known in the art. The physical caches requires address translation for data access from cache and hence cache translating virtual to physical addressing is known in the art, and examiner takes official notice of that. Physical caches are simpler to build; hence it would have been

Art Unit: 2188

obvious to one having ordinary skill in the art at the time of the invention to use physical cache performing address translation in the system of Zaidi, Jeddeloh and Loafman.

As per claim 33, Loafman teaches loading multiple lines of data from secondary storage device to cache as explained with respect to claim 1 above.

10. Claims 20-24 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jeddeloh and Loafman as applied to claim 18 above, and further in view of Alexander et al. (US 6,131,155).

As per claims 20 and 23, Zaidi, Jeddeloh and Loafman teach all the limitations of independent claim 18, but fail to teach direct interface to bypass cache. Alexander teaches a CPU programmed to bypass a cache when necessary (Alexander, abstract).

It would have been obvious to one having ordinary skill in the art at the time of the invention to utilize the direct access interface to memory, bypassing the cache as taught by Alexander in the system of Zaidi, Jeddeloh and Loafman, because data caches provides performance improvement only if program execution performs repeated accesses of data over a short period of time to a small group of data and large amounts of data transfers degrades the performance by the phenomenon known as thrashing, so bypassing a cache and directly reading burst data from memory increases the performance (Alexander, abstract, col. 2, lines 21-56). Also providing a direct/exclusive and independent interface avoids bus or memory bank conflict as explained with respect to claims 1, 4, 18 and 26 above.

As per limitation of claims 21-22 and 24, Alexander teaches that when necessary in order to fetch or store (reading from and writing/programming to memory device) data items directly from/to the memory, ensuring data accesses exhibiting a high degree of locality are made to the cache, while those accesses that are non-local are made directly to the main memory, bypassing the cache (Alexander, abstract), proves that depending upon addresses certain access are only performed through the direct memory interface, bypassing the cache.

11. Claims 28-32 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jeddeloh, Loafman and Barroso as applied to claim 26 above, and further in view of Alexander et al. (US 6,131,155).

As per claim 28, Zaidi, Jeddeloh, Loafman and Barroso teach all the limitations of independent claim 26, but fail to teach direct interface to bypass cache. Alexander teaches a CPU programmed to bypass a cache when necessary (Alexander, abstract).

It would have been obvious to one having ordinary skill in the art at the time of the invention to utilize the direct access interface to memory, bypassing the cache as taught by Alexander in the system of Zaidi, Jeddeloh, Loafman and Barroso, because data caches provides performance improvement only if program execution performs repeated accesses of data over a short period of time to a small group of data and large amounts of data transfers degrades the performance by the phenomenon known as thrashing, so bypassing a cache and directly reading burst data from memory increases the performance (Alexander, abstract, col. 2, lines 21-56). Also providing a

direct/exclusive and independent interface avoids bus or memory bank conflict as explained with respect to claims 1, 4, 18 and 26 above.

As per limitation of claims 29 and 30-32, Alexander teaches that when necessary in order to fetch or store (reading from and writing/programming to memory device) data items directly from/to the memory, ensuring data accesses exhibiting a high degree of locality are made to the cache, while those accesses that are non-local are made directly to the main memory, bypassing the cache (Alexander, abstract), proves that depending upon addresses certain access are only performed through the direct memory interface, bypassing the cache.

As per limitation of claim 31, Zaidi and Barroso teach an arbiter and MAC (Zaidi, fig. 2, items 242, 244) and switch (Barroso, fig.1, item 120) to resolve memory access conflict (Zaidi, column 23, lines 40-45) but fail to teach arbiter for direct read/write interface. It would have been obvious to one having ordinary skill in the art at the time of the invention would provide arbiter for direct (bypassing cache interface) interface, because when multiple CPUs accessing memory device, providing arbitration avoids the conflict for same data.

### ***Conclusion***

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.


13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kaushikkumar Patel whose telephone number is 571-272-5536. The examiner can normally be reached on 8.00 am - 4.30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


Kaushikkumar Patel

Art Unit: 2188



kmp

Examiner  
Art Unit 2188



HYUNG SUOH  
SUPERVISORY PATENT EXAMINER

4-29-07